Comparison of Different Evolutionary Methodologies Applied to Electronic Filter Design

Ricardo S. Zebulum [1] Marco Aurelio Pacheco [2] Marley Vellasco [2]

1- Centre for Computational Neurosciences and Robotics, University of Sussex Brighton BN1 9SB, UK, E-mail: ricardoz@cogs.susx.ac.uk
2 - Centro de Inteligencia Computacional Aplicada Pontificia Universidade Catolica, Rio de Janeiro - Brasil

Abstract

We present in this work the application of a set of different evolutionary methodologies in the problem of electronic filter design. The main objectives are to find out which constraints in the filter topologies, if any, must be observed along the evolutionary process and to study the problem of convergence to parsimonious circuits. The new area of Evolutionary Electronics is introduced, an evolutionary methodology based on variable length representation is presented and the results on the evolution of low-pass and band-pass filters are described.

1. Introduction

This work focuses on the application of evolutionary systems [4] in engineering design. Particularly, the application of evolutionary techniques in the area of electronic design and optimisation gave birth to a new and promising area of research, Evolutionary Electronics[13][6]. The aim of this area is the creation of new automation design techniques for electronic circuits, based on the Darwinian concepts of natural selection, recombination and mutation. We present and apply in this work evolutionary methodologies based on fixed and variable length genotypes in the problem of passive analog filter design.

This work is divided into six sections; section 2 introduces the basic concepts of Evolutionary Electronics; section 3 presents some previous works in the area of passive analog filter design; section 4 presents our evolutionary methodologies; section 5 describes a series of experiments made by the authors in electronic filters design; we compare the results for different filter specifications (low-pass and band-pass) and also for different evolutionary frameworks, i.e., using fixed or variable length representations. Section 6 concludes this work.

2. Evolutionary Electronics

Evolutionary Electronics extends the concepts of Genetic Algorithms to the evolution of electronic circuits. The main idea behind Evolutionary Electronics is that each possible electronic circuit can be represented as an individual or a chromosome of an evolutionary process, which performs standard genetic operations over the circuits, such as selection, crossover and mutation [13]. We may visualise the task of Evolutionary Electronics as the one of sampling a search space S consisting of electronic circuits [3]. The sampled search space S has the following main features:

1. It is usually very large and formed by circuits of different sizes and topologies;

2. It can be visualised as formed of two subspaces [3]: the sub-space of the compliant circuits, S_c, and the sub-space of the non-compliant circuits, S_{nc}.

The main idea expressed by the last feature is that, given a system specification or target, the search space will be usually formed by circuits that satisfy the user specifications, contained in S_c, and circuits that do not satisfy the required specifications, contained in S_{nc}. The Artificial Evolutionary System should converge to a solution in S_c. Nevertheless, engineers are not interested in any solution of S_c, but only in those solutions which are optimal or quasi-optimal in terms of one or more specific criteria, such as the number of circuit components, area, speed or its power consumption. Therefore, the Artificial Evolutionary System must converge to a design contained in the restricted subspace of the engineering solutions, S_e, where $S_e \subset S_c$. This can be viewed in Figure 1, which shows an example of the mapping between the fitness landscape and the Evolutionary Electronics search space subsets, as well as an exam-

0-7803-4869-9/98 $10.00©1998 IEEE

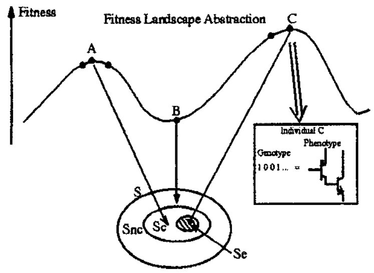

Figure 1 - Evolutionary Electronics Genome Space

ple of the genotype - phenotype mapping in Evolutionary Electronics; in this figure, $A \in S_c$, $B \in S_{nc}$ and $C \in S_e$ (shaded area). The individual C, with the highest fitness, suits as an engineering solution.

Due to the broad scope of the area, researchers have been focusing on different problems, such as placement and routing, evolution of digital circuits based on Boolean gates [12], evolution of sequential circuits [5][12], evolution of passive and active analog circuits [9][7] [2], evolution of operational amplifiers [11][1] and transistor size optimisation. Of great relevance are the works focusing on intrinsic hardware evolution [13], in which evolution is performed in the own silicon, exploring, therefore, all the physical properties of the medium. This particular area is called Evolvable Hardware [5].

3. Previous Works

We are particularly addressing the case of evolving analog circuits based on resistors, capacitors and inductors. This task has already been tackled from different evolutionary perspectives by other authors recently.

J.B. Grimbleby, from the University of Reading [2], has applied Evolutionary Algorithms (EAs) to synthesise filters for particular frequency and time domain specifications. In his approach, the EAs are used to find the network topology; the component values are determined by numerical optimisation, without evolution. He has used fixed length string structures and integer number chromosome representation.

In another application, Horrocks and Spittle, from the University of Wales [7], have used Genetic Algorithms to deal with the problem of preferred values electric components selection, particularly resistors and capacitors, to implement active filters. In [8], Horrocks and Khalifa extend this study, taking into account parasitic effects in resistors, inductors and capacitors. They evolved a low pass filter with 9 components encoded in 9 genes through binary representation. Though, they have not addressed the case of searching for new topologies.

Using a different evolutionary methodology, Koza et. al. have made a large number of works applying the Genetic Programming (GP) methodology in electronic circuit synthesis. In[9][10][1], they have applied Genetic Programming and the Circuit Constructing Tree methodologies[9] to the evolution of a Crossover Filter, evolution of a Brick-Wall Low Pass Filter, evolution of an Asymmetric Bandpass Filter and evolution of an operational amplifier. Both the filter topologies and the component values have been evolved and the SPICE simulator has been used to evaluate the circuits.

4. Evolutionary Methodologies

Based on these previous works and on the main features needed for analog design automation tools, we devised three main issues:

1. Evolution of the circuit topologies;
2. Evolution of the components nature and values;
3. The need to seek for parsimonious solutions.

The first issue is examined by comparing two experiments (see next section): one in which the topologies sampled by the Evolutionary Algorithm (EA) are constrained to be arranged into parallel meshes and another in which this constraint is removed and the EAs are able to make any component arrangement.

The second issue raised is examined by allowing the EA to choose the nature and the value of the components; however, in order to make this application as near as possible to the real electronics world, we restrict resistor, capacitors and inductors to take only the so-called preferred values [7], that are more commonly manufactured.

The third issue is of major importance for our work, since it is directly related to our evolutionary methodology. If we are to use genotypes represented by strings of fixed length to this problem, we have to guess, possibly using our previous knowledge, the number of components that are necessary to satisfy a particular filter design specification. Furthermore, penalties will have to be applied in the fitness function for large circuits, if parsimony is also an important objective. On the other hand, when using GP, care should be taken to control the depth size of the trees.

In order to overcome the problems described above, we decided to use an EA in which circuits are represented by strings that are allowed to grow along the evolutionary process. We call it the Increasing

Length Genotypes Approach, which is based on the ideas of the Species Adaptation Genetic Algorithm (SAGA) [4]; all electronic circuits of the initial population will have a small number of components, which increases gradually along the evolutionary process. Therefore, the EA starts sampling subspaces of parsimonious solutions, going toward subspaces of more complex solutions as long as design requirements are not satisfied. The selection pressure provides a way to stop the growing process when circuits close to the specification are found, although care must be taken with local optima.

Therefore, in addition to crossover and mutation, we have also used the Increasing Length Operator, which increases, with a low probability [4], the length of each genotype by one gene; each new gene created by this operator is randomly initialised. Particularly, our crossover operator did not require maximisation of similarity between genotype segments, which is a concept used in the SAGA framework. The mutation operator actuates over integer values by changing them, with equal probability, to any other value within the alphabet used. Tests have been made with both fixed and adaptive mutation rates. In the latter, the mutation rate increases as the average genotype length grows.

5. Case Studies

We have devised three sets of tests to evaluate different evolutionary methodologies:

1. Variable length representation EAs, constraining the topologies of the filters to meshes;
2. Fixed length representation EAs, making no constraints to the filter topologies;
3. Variable length representation EAs, making no constraints to the filter topologies.

In order to evaluate the above evolutionary methodologies, we have used two test cases, a low-pass and a band-pass filter, with the following frequency requirements:

1. Low-pass filter with passing band up to 1000 Hz and stop band above 2000 Hz [10];
2. Band-pass filter with passing band between 2000 Hz and 3000 Hz and stop band below 1000 Hz and above 4000 Hz.

Based on Koza's previous works [9][10], we have devised the following fitness evaluation function:

$$\text{fitness} = \sum_{i=1}^{n} w_i |Target_i - Output_i| \qquad (1)$$

According to equation 1, the fitness is the weighted sum of the deviations between the frequency response specification, $Target_i$ and the frequency response obtained $Output_i$. The frequency axis is uniformly sampled over the band of interest, usually ranging from 0 to 10 kHz. The weight w_i takes a maximum value for frequency points inside the passband, an intermediate value for frequency points inside the stop band and a minimum value for other frequency points. We set w_i equal to 20, 10 and 1 for these three cases respectively. Since we are using the roulette well selection method, the value given by equation 1 had to be inverted. Although this fitness equation worked well, setting the values of the weights has been an interactive and time consuming process.

In order to evaluate the circuits, we have used, according to the case, a hand written simulator, the SMASH simulator, from Dolphin corporation, and the SPICE simulator.

5.1 First Test Set

In the first set we constrained the circuit topologies to be made up of parallel meshes, each one having two components. Each gene will encode a mesh of the circuit, defining the nature of the two components and their values (chosen from the preferred manufactured values). Therefore, each time a genotype is increased by one (random) gene, one mesh is added to the circuit. The initial population consists of circuits of only one or two meshes and may grow until five meshes. Figures 2 and 3 show the evolved circuits for the low-pass and band-pass specifications respectively and Figures 4 and 5 show their respective frequency responses (output voltage in decibels versus frequency in Hertz).

We have used a population of 50 individuals, running along 500 generations, crossover rate of 0.7, adaptive mutation rate, and increasing length operator rate of 0.05. Fitness proportional selection and one-point crossover have been used. Each genotype is formed of up to 30 positions (5 genes made up of 6 loci). An integer number representation has been used [2] and, according to this representation, each gene may represent a total of 82,944 meshes (different combinations of capacitors, resistors and inductors of different values). A hand-written circuit simulator based on the Laplace analysis has been used.

5.2 Second Test Set

In the second test set we have used fixed length representation, though allowing any kind of connections among the components. Each gene in the genotype now describes a single component, by determin-

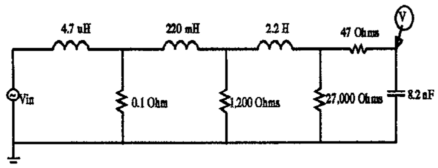

Figure 2 - Evolved Low Pass Filter (First Set)

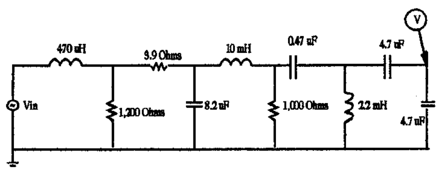

Figure 3 - Evolved Band Pass Filter (First Set)

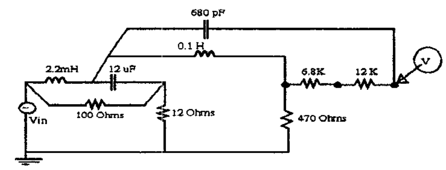

Figure 6 - Evolved Low Pass Filter (Second Set)

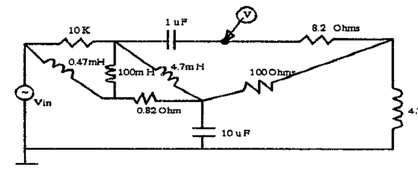

Figure 7 - Evolved Band Pass Filter (Second Set)

ing its nature (resistor, capacitor or inductor), value (from the manufactured preferred values) and connections points. We established that all genotypes would be made up of 10 genes, i. e., each circuit with 10 components. The SMASH simulator has been used to evaluate the circuits, in the AC analysis mode. As any kind of arrangement among the components is now possible, many topologies are now not simulatable, receiving a negative fitness value of -1000 [10].

We have used an Evolutionary Algorithm with 40 individuals, running over 200 generations, crossover rate of 0.7, mutation rate of 0.02 per genotype position, linear rank selection with truncation, and uniform crossover. Each genotype is formed by 50 positions (10 genes made up of 5 loci). An integer number representation has been used and, according to this representation, each gene may represent a total of 14,112 components (different values of capacitors, resistors and inductors).

Figures 6 and 7 show the evolved circuits for the low-pass and band-pass specifications respectively and Figures 8 and 9 show their respective frequency responses. These circuits have also been simulated in SPICE, showing the same results. Note that the low-pass filter has only 9 components, because one component was found to have no effect in the circuit behaviour.

5.3 Third Test Set

In the third test set we have allowed any topology arrangement between components, using a variable size representation. Each gene represents a particular component, in the way described in the last section. We initialised all the individuals in the population with five random genes and applied an especial operator to increase the genotypes along the evolutionary process. Opposing to the first set of ex-

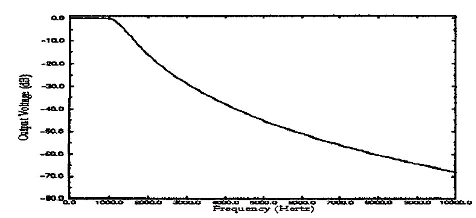

Figure 4 - Frequency Response for the Circuit of Fig. 2

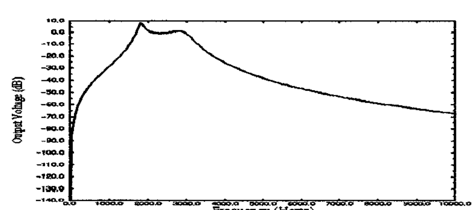

Figure 5 - Frequency Response for the Circuit of Fig. 3

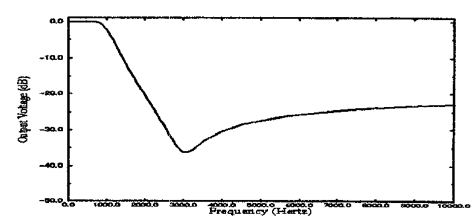

Figure 8 - Frequency Response for the Circuit of Fig. 6

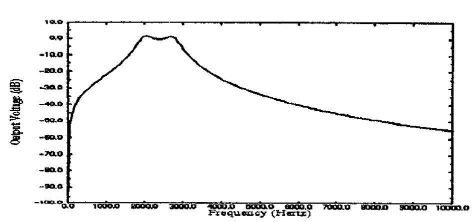

Figure 9 - Frequency Response for the Circuit of Fig. 7

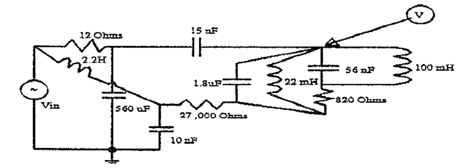

Figure 10 - Evolved Low Pass Filter (Third Set)

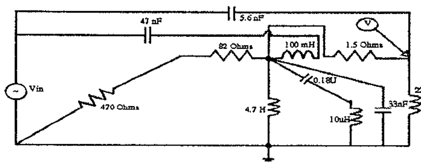

Figure 11 - Evolved Band Pass Filter (Third Set)

periments, in which the unit of increase was a mesh made up of two components, in this case the unit of increase was one component. The SMASH simulator has been used to evaluate the circuits in the AC analysis mode.

We have used a population of 40 individuals, running along 300 generations, crossover rate of 0.7, mutation rate of 0.02 per genotype position and increasing length operator rate of 0.1. Linear rank selection with truncation, uniform crossover and integer number representation have been used. The genotypes are initialised with five genes, each gene being made up of 5 positions and being able to express a total of 14,112 components. They are allowed to grow up to a total of 15 genes.

Figures 10 and 11 show the evolved circuits for the low-pass and band-pass specifications respectively and Figures 12 and 13 show their respective frequency responses. These circuits have also been simulated in SPICE, showing the same results.

Figures 14 and 15 show the average genotype lengths, in terms of number of electric components, of the population along the evolutionary process for particular runs on the low-pass and band-pass cases respectively. It can be seen from the figures that there is a tendency for the genotypes to grow along the evolutionary process, due to the application of the Increasing Length Operator. Although the Increasing Length operator is applied at a constant rate along the evolutionary process, we can verify from the figures that there are periods in which the average length of the genotypes stays stable. In these periods, the EA is failing to find fitter solutions in less parsimonious subspaces; so, selection pressure keeps the average length of the population constant. This fact suggests that our approach is capable of finding more parsimonious solutions.

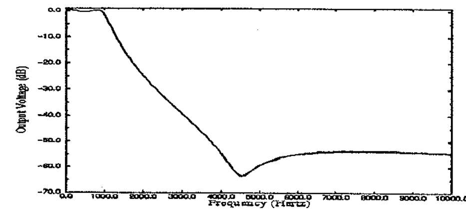

Figure 12 - Frequency Response for the Circuit of Fig. 10

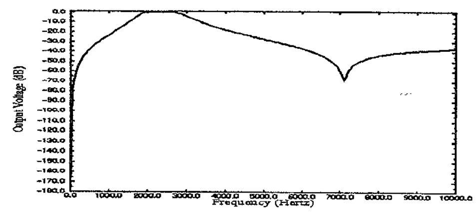

Figure 13 - Frequency Response for the Circuit of Fig. 11

5.4 Summary of the Results

For the sake of comparison, Tables I and II summarise the results obtained over the experiments performed. In these tables, NC is the number of components of the best solution and A(1K), A(2K), A(3K) and A(4K) are the response of the circuits, in dB, for the frequencies of interest, i. e., 1 kHz, 2 kHz, 3 kHz and 4 kHz. For the low pass filter, good designs should have high A(1K) values and low A(2K) values; for the band-pass filter, good designs should have high A(2K) and A(3K) values and low A(1K) and A(4K) values. The lines labelled 1, 2 and 3 accounts for the first, second and third set of experiments respectively. It can be seen from these tables that, whereas the three approaches perform similarly in both cases, the first approach arrived at

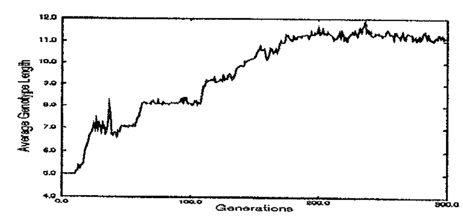

Figure 14 - Average genotypes length (lowpass filter)

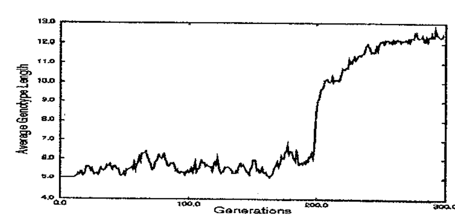

Figure 15 - Average genotypes length (bandpass filter)

TABLE I
SUMMARY ON LOW-PASS FILTER EVOLUTION

-	NC	A(1K)	A(2K)
1	8	-0.1	-16.1
2	9	-2.3	-20.3
3	11	-1.6	-24.8

TABLE II
SUMMARY ON BAND-PASS FILTER EVOLUTION

-	NC	A(1K)	A(2K)	A(3K)	A(4K)
1	10	-22.3	7.7	4.98	-19.4
2	10	-22.2	1.5	-5.9	-24.5
3	11	-23.3	0.2	-3.7	-17.8

a smaller circuit in the first case.

Comparing the performance of this evolutionary methodology with the genetic programming methodology in the evolution of the low-pass filter [10], it has been verified that this methodology arrived at circuits with similar frequency responses, though processing fewer individuals than in the genetic programming methodology.

In terms of time considerations, the first set of experiments are also faster, because we could write a simulator in C to evaluate the circuits, since they are made up of simple meshes. For the sake of comparison, the written simulator based on Laplace analysis takes around 1 second to evaluate a circuit, while standard simulators used in the second and third sets take around 3 seconds to do the same.

Even though the solutions obtained in the three sets of experiments are satisfactory in terms of performance, a problem remains to be addressed: the circuits obtained in all the tests have presented large inductor values, which may turn difficult the use of these circuits in real electronic applications. Although the first methodology has outperformed the other ones when time is taken into account, the issue of component size minimisation may change the situation. When tuning the fitness evaluation function to search also for small sized components, it is likely that the new topologies, potentially found in the second and third sets, may be more successful.

6. Conclusions

Different evolutionary approaches have been applied to the problem of passive electronic filter design. The issues of variable and fixed EA representation and of imposing constraints in the circuit topologies have been investigated. We are now studying the problem of evolution of topologies with small sized components, in order to make this tool suitable to be applied in integrated circuits design.

7. Acknowledgements

The authors would like to thank CAPES, Brazilian Federal Agency, for sponsoring this work.

REFERENCES

[1] Bennett III, F. H., Koza, J. R., Andre, D., Keane, M. A., " Evolution of a 60 Decibel Op Amp Using Genetic Programming", Proceedings of the First International Conference on Evolvable Systems (ICES96), LNCS 1259, pp. 455-469, Tsukuba, Japan, October, 1996.

[2] Grimbleby, J. B. , "Automatic Analogue Network Synthesis Using Genetic Algorithms", Proceedings of the First IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems (GALESIAS - 95), pp.53-58, UK, 1995.

[3] Harvey, I., Thompson, A., "Through the Labyrinth Evolution Finds a Way: A Silicon Ridge", Proceedings of the First International Conference on Evolvable Systems (ICES96), LNCS 1259, pp. 406-422, Tsukuba, Japan, October, 1996.

[4] Harvey, I., "The Artificial Evolution of Adaptive Behaviour", PhD Thesis, University of Sussex, School of Cognitive and Computing Sciences, September, 1993.

[5] Higuchi, T., Iba, H., Manderick, B., "Evolvable Hardware", in Massively Parallel Artificial Intelligence (ed. H. Kitano), MIT Press, 1994.

[6] Hirst, T. , "Evolutionary Signal Processing: A Preliminary Report", Proceedings of the Fourth Conference on Artificial Life, pp. 425-431, Brighton, UK, July, 1997.

[7] Horrocks, D. H, Spittle, M. C., "Component Value Selection For Activate Filters Using Genetic Algorithms", 1st On-line Workshop on Soft Computing (WSC1), Special Session on Evolutionary Electronics, August 19-30, 1996.

[8] Horrocks, D. H., Khalifa, Y. M. A., "Genetic Algorithm Design of Electronic Analogue Circuits Including Parasitic Effects", 1st On-line Workshop on Soft Computing (WSC1), Special Session on Evolutionary Electronics, August 19-30, 1996.

[9] Koza J. R., Bennett III F. H., Andre, D. , Keane, M. A., "Four Problems for which a Computer Program Evolved By Genetic Programming is Competitive with Human Performance", Proc. Of 1996 IEEE International Conference on Evolutionary Computation, IEEE Press., Pages 1-10.

[10] Koza J. R., Bennett III F. H., Andre, D. , Keane, M. A, "Toward Evolution of Electronic Animals Using Genetic Programming", Artificial Life V: Proceedings of the Fifth International Workshop on the Synthesis and Simulation of Living Systems. Cambridge, MA: The MIT Press, 1996.

[11] Kruiskamp, W. and Leenaerts, D., "DARWIN: CMOS opamp synthesis by means of genetic algorithm", Proceedings of the 32nd Design Automation Conference, New York, NY: Association for Computer Machinery, pp. 698-703, 1995.

[12] Miller, J. F., Thomson, P. , "Combinational and Sequential Logic Optimization Using Genetic Algorithms", Proceedings of the First IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems (GALESIA - 95), pp.34-38, UK, 1995.

[13] Thompson, A., "An Evolved Circuit, Intrinsic in Silicon, entwined with physics", Proceedings of the First International Conference on Evolvable Systems (ICES96), Tsukuba, LNCS 1259, pp. 390-405, Japan, October, 1996.